

**PENGEMBANGAN SISI BACK-END APLIKASI PENGELOLAAN  
SURAT MENGGUNAKAN FRAMEWORK DJANGO DI  
UNIVERSITAS MUHAMMADIYAH SURAKARTA**



**Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada Jurusan  
Informatika Fakultas Komunikasi dan Informatika**

**Oleh:**

**Muhammad Hammam Islami**

**L200160140**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2020**

HALAMAN PERSETUJUAN

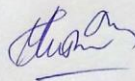
**PENGEMBANGAN SISI BACK-END APLIKASI PENGELOLAAN  
SURAT MENGGUNAKAN FRAMEWORK DJANGO DI UNIVERSITAS  
MUHAMMADIYAH SURAKARTA**

**PUBLIKASI ILMIAH**

oleh:

**MUHAMMAD HAMMAM ISLAMI**  
**L200160140**

Telah diperiksa dan disetujui untuk diuji oleh:  
Dosen Pembimbing



**Husni Thamrin, M.T, Ph.D**  
**NIK.706**

## HALAMAN PENGESAHAN

### PENGEMBANGAN SISI BACK-END APLIKASI PENGELOLAAN SURAT MENGGUNAKAN FRAMEWORK DJANGO DI UNIVERSITAS MUHAMMADIYAH SURAKARTA

OLEH

MUHAMMAD HAMMAM ISLAMI

L200160140

Telah dipertahankan di depan Dewan Penguji  
Fakultas Komunikasi dan Informatika  
Universitas Muhammadiyah Surakarta  
Pada hari Sabtu, 7 Maret 2020  
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. Husni Thamrin, S.T., M.T., Ph.D.  
(Ketua Dewan Penguji)
2. Heru Supriyono, S.T., M.Sc., Ph.D.  
(Anggota I Dewan Penguji)
3. Dedi Gunawan, S.T., M.Sc., Ph.D.  
(Anggota II Dewan Penguji)

(.....)  
(.....)  
(.....)



Dekan  
Fakultas Komunikasi dan Informatika

Dedi Gunawan, S.T., M.Sc., Ph.D.  
881



Ketua  
Jurusan Studi Informatika

Heru Supriyono, S.T., M.Sc., Ph.D.  
970

## **PERNYATAAN**

Dengan ini saya menyatakan bahwa dalam publikasi ilmiah ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

**Surakarta, 7 Maret 2020**

Penulis



**MUHAMMAD HAMMAM ISLAMI**

**L200160140**





**UNIVERSITAS MUHAMMADIYAH SURAKARTA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
PROGRAM STUDI INFORMATIKA**

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448  
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: [informatika@ums.ac.id](mailto:informatika@ums.ac.id)

**SURAT KETERANGAN LULUS PLAGIASI**

**No Surat 143/A.3-IL3/inf-FKI/IV/2020**

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama : Muhammad Hammam Islami  
NIM : L200160140  
Judul : **Pengembangan Sisi Back-End Aplikasi Pengelolaan Surat  
Menggunakan Framework Django di Universitas Muhammadiyah  
Surakarta**  
Program Studi : Informatika  
Status : Lulus

Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Surakarta, 24 April 2020  
Biro Skripsi Informatika

**Ihsan Cahyo Utomo, S.Kom., M.Kom.**



# **PENGEMBANGAN SISI BACK-END APLIKASI PENGELOLAAN SURAT MENGUNAKAN FRAMEWORK DJANGO DI UNIVERSITAS MUHAMMADIYAH SURAKARTA**

## **Abstrak**

Perkembangan teknologi dan internet mengakibatkan munculnya digitalisasi di banyak lini aktifitas manusia. Salah satunya adalah pengiriman surat resmi. Surat resmi yang dulunya dikirim dalam bentuk cetak, kini dapat dikirimkan dalam bentuk digital. Hal ini memudahkan pengiriman surat dan efisiensi waktu. Oleh karena itu, untuk memudahkan penyuratan perlu adanya digitalisasi surat-menyurat resmi di dalam sebuah instansi. Di Universitas Muhammadiyah Surakarta sudah terdapat aplikasi pengelolaan surat resmi. Namun, aplikasi tersebut belum dapat mengirimkan surat antar instansi dan masih terbatas untuk pembuatan surat dan penyimpanan surat dalam instansi. Oleh karena itu perlu adanya pengembangan sistem agar antar instansi di UMS dapat saling mengirimkan surat. Dalam proyek ini, sistem yang akan dibangun adalah sebuah *RESTful API* pengelolaan surat UMS. *RESTful API* dipilih, karena memiliki fleksibilitas dalam hal implementasi sistem dan perkembangan teknologi *website* yang menuju pada *Single Page Application*. Sedangkan *framework* yang digunakan adalah *Django Rest Framework*. Penggunaan *Django Rest Framework* dikarenakan kekuatan dan fleksibilitasnya untuk membangun *RESTful API*. Hasil dari proyek ini adalah alamat-alamat *URL* yang dapat dikonsumsi untuk mengelola surat-menyurat di UMS.

**Kata Kunci:** pengelolaan surat-menyurat, *restful api*, *django rest framework*.

## **Abstract**

The development of technology and the internet result in the emergence of digitalization in many aspects of human activity. One of them is official letters submission. Official letters that were sent in print can now be sent in digital form. This makes it easy to send letters and leads to time efficiency. Therefore, to facilitate correspondence, it is necessary to digitize official correspondence within an unit. At Muhammadiyah University of Surakarta there is already an official letter management application. However, the application has not been able to send letters between agencies and is still limited to making letters and storing letters inside an unit. Therefore it is necessary to develop a system so that between units at UMS can send each other letters. In this project, the system is built on *RESTful API*. *RESTful API* was chosen, because it has flexibility in terms of system implementation and website technology development that leads to *Single Page Application*. While the framework used is the *Django Rest Framework*. The use of the *Django Rest Framework* is due to its strength and flexibility to build a *RESTful API*. The results of this project are *URL* addresses that can be consumed for managing correspondence at UMS.

**Keywords:** mail management, *restful api*, *django rest framework*

## 1. PENDAHULUAN

Surat merupakan sarana komunikasi antar dua belah pihak yang bertujuan memberikan informasi dari pengirim surat ke penerima surat. Surat memiliki banyak jenis, salah satu diantaranya adalah surat resmi. Surat resmi adalah surat yang dipergunakan untuk kepentingan bersifat resmi, baik yang ditulis dari perseorangan, instansi, lembaga, maupun organisasi (Supriyana, Azmin, Nurriyani, & Rahmawati, 2015). Formalitas di dalam surat diperlukan, agar isi surat dapat dipercaya dan dipertanggungjawabkan. Selain itu surat resmi juga digunakan sebagai bukti tertulis (otentik), bukti historis, dan bukti kronologis.

Di dalam sebuah instansi, surat resmi sering digunakan. Surat resmi memiliki gaya penulisan masing-masing tergantung dari jenis surat resmi itu sendiri. Bahkan dalam surat resmi, format penulisan surat pun juga menjadi salah satu tolok ukur yang penting (Supriyana et al., 2015). Beberapa surat resmi di instansi diantaranya adalah surat keputusan, surat permohonan, surat kuasa, surat perintah, surat panggilan, dan surat undangan.

Dengan banyaknya surat resmi yang digunakan, jika tidak dikelola dengan baik, akan menimbulkan berbagai masalah. Seperti pencarian surat yang sulit jika sewaktu-waktu surat ingin dicari, kemungkinan penomoran ganda yang semakin tinggi, atau kemungkinan surat terselip. Oleh karena itu, diperlukan sebuah solusi agar pengelolaan surat dapat terorganisir dengan baik. Salah satu hal yang dapat dilakukan adalah dengan mendigitalisasi kegiatan surat menyurat.

Menurut literatur, digitalisasi, atau transformasi digital, mengacu pada “perubahan yang terkait dengan penerapan teknologi digital di semua aspek masyarakat manusia” (Parviainen, Tihinen, Kääriäinen, & Teppola, 2017). Digitalisasi sudah menjadi salah satu dari transformasi berkelanjutan paling signifikan pada masyarakat saat ini dan mencakup banyak element mulai dari bisnis sampai kehidupan sehari-hari (Hagberg, Sundstrom, & Nicklas, 2016). Dengan mendigitalisasi kegiatan surat-menyurat, permasalahan seperti penomoran ganda, penumpukan surat dan pencarian surat yang sulit akan teratasi karena sudah terdapat sistem yang mengelola surat. Hal ini juga dapat mengurangi pembuatan surat cetak dan beralih ke surat *paperless*.

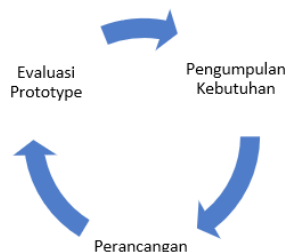
Di Universitas Muhammadiyah Surakarta, digitalisasi pada surat resmi sudah direalisasikan. Rektorat dan beberapa fakultas memiliki aplikasi pengelolaan surat



masing-masing yang terdiri dari surat masuk, surat keluar, dan surat keputusan. Akan tetapi, aplikasi pengelolaan surat tersebut, belum terhubung satu sama lain. Sehingga pengiriman surat resmi dari rektorat atau antar fakultas masih dikirim manual. Oleh karena itu, diperlukan aplikasi pengelolaan surat yang saling terintegrasi. Dengannya, diharapkan dapat memudahkan pengiriman surat dari rektorat, antar fakultas, atau antar jabatan. Sehingga perkembangan teknologi tersebut, diharap dapat mempercepat pengiriman surat dan mencegah surat tersebut tidak sampai pada tujuan.

## 2. METODE

Metode yang digunakan untuk penelitian ini adalah metode *prototype*. Metode *prototype* dipilih, karena dengan metode ini penulis bisa berkomunikasi dengan klien secara lebih intensif sehingga sistem yang dibuat sesuai dengan keinginan klien. Metode *prototype* membantu mendapatkan umpan balik yang berharga dari klien dan membantu *designer* dan *developer* memahami tentang apa sebenarnya yang diharapkan dari produk yang sedang dikembangkan (Iqbal & Idrees, 2017) . Dengan menggunakan metode *prototype*, proyek pengembangan dapat dengan mudah beradaptasi dengan perubahan permintaan, karena ada umpan balik yang konstan (Isaias & Issa, 2015). Metode *prototype* bisa digambarkan pada gambar 1 (Herlambang & Thamrin, 2018).



Gambar 1. Metode *prototype*

### 2.1. Pengumpulan Kebutuhan

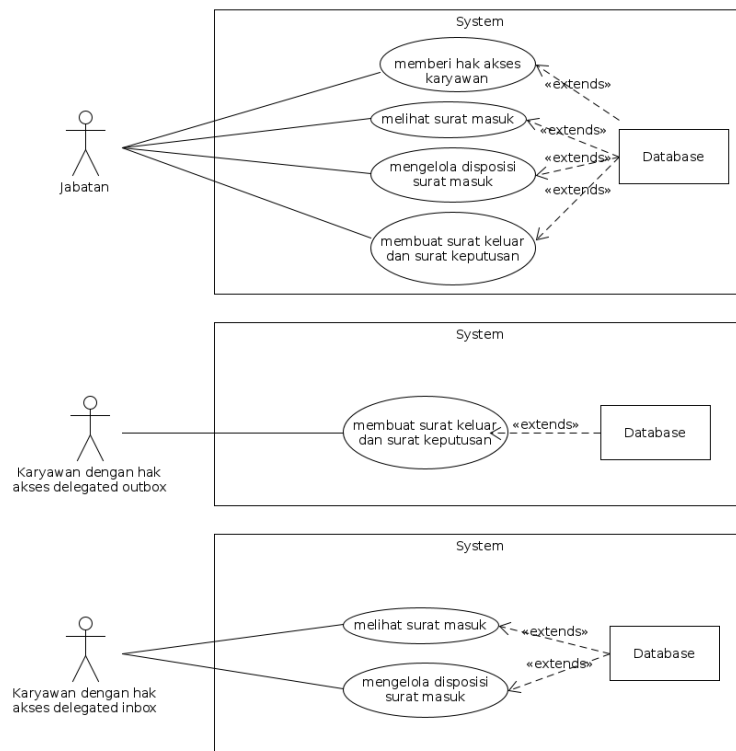
Pada tahap ini, penulis bertanya kepada *developer* aplikasi pengelolaan surat Universitas Muhammadiyah Surakarta terkait aplikasi pengelolaan surat yang sudah dibuat. Dari wawancara didapatkan bahwa aplikasi pengelolaan surat tersebut belum terintegrasi satu sama lain, sehingga perlu dikembangkan aplikasi pengelolaan surat yang saling terintegrasi dalam lingkup universitas agar rektorat atau antar fakultas dapat mengirim surat dengan lebih cepat dan mudah. Dari wawancara juga didapatkan

pembagian surat dibagi menjadi 3, yaitu surat masuk, surat keluar, dan surat keputusan. Surat tersebut hanya dapat dilihat dan dikelola oleh pejabat di lembaga pemilik surat atau karyawan yang didelegasikan untuk mengelola surat.

## **2.2. Perancangan**

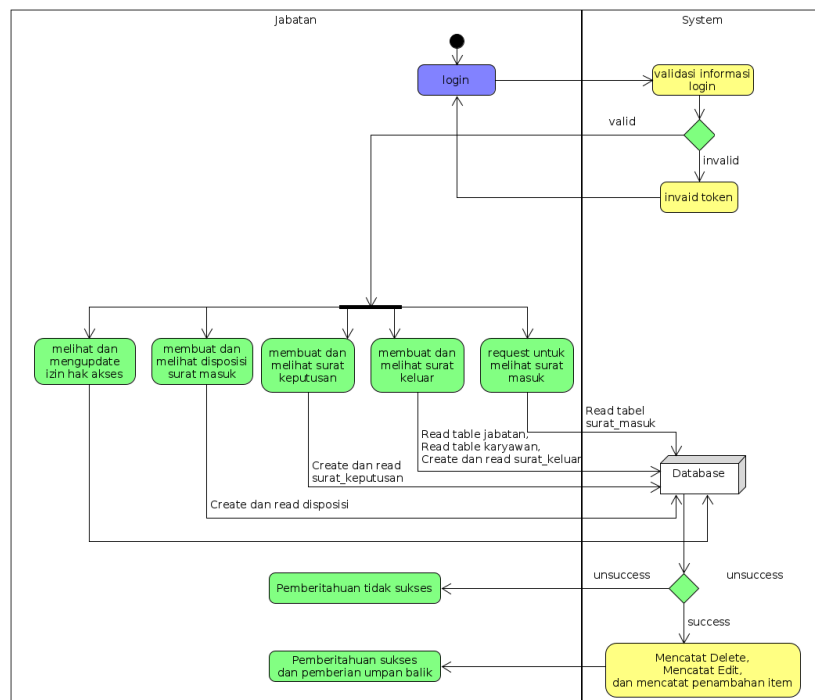
Pada penelitian yang dilakukan, sistem yang akan dibuat adalah sebuah *RESTful API* menggunakan *Django REST framework*. *Django REST framework* merupakan *toolkit* khusus milik *framework django* yang digunakan untuk membangun *web API*. Alasan penggunaan *django* karena *django* merupakan *framework* yang sangat terstruktur dan serbaguna. *Framework django* terstruktur sedemikian rupa sehingga sangat modular dan serbaguna (Adithela, Christie, Marru, & Pierce, 2017).

Ada 2 *user* yang dapat mengakses *API* surat-menyurat ini, yaitu jabatan dan karyawan. Jabatan dapat melihat surat masuk, mengelola disposisi surat masuk, membuat surat keluar dan surat keputusan, serta memberi hak akses delegasi kepada karyawan sesuai dengan jabatan yang *user* miliki. Sedangkan karyawan tidak memiliki kewenangan dalam mengelola surat kecuali diberi hak akses delegasi. Hak akses delegasi terdiri dari 2, *delegated inbox* dan *delegated outbox*. *Delegated inbox* memberi hak akses karyawan untuk melihat surat masuk dan membuat disposisi surat masuk, sedangkan *delegated outbox* memberi hak akses karyawan untuk membuat surat keluar dan surat keputusan. *Use case diagram* untuk *API* surat-menyurat ini dapat dilihat pada Gambar 2. *Use case diagram* digunakan untuk merepresentasikan interaksi *user* dengan sistem yang menunjukkan hubungan antara *user* dan berbagai kasus penggunaan dimana *user* terlibat.

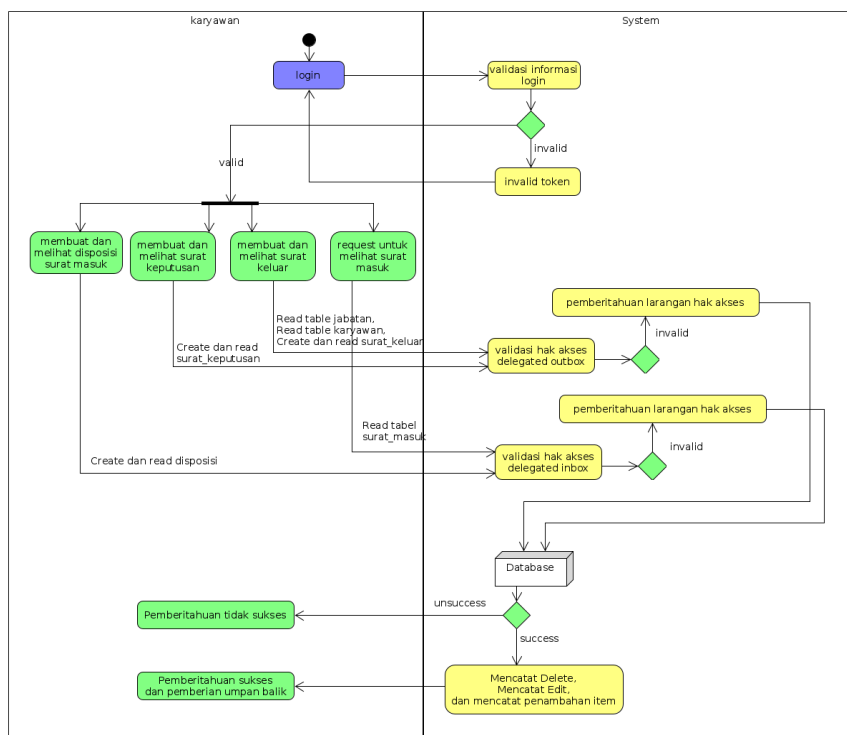


Gambar 2. Use case diagram API aplikasi pengelolaan surat

Pada Gambar 3 dan 4 menjelaskan aksi-aksi yang dapat dilakukan oleh para aktor yang telah disebutkan pada *use case diagram*. *Activity diagram* tersebut memperlihatkan proses dari login sampai aktor mendapat umpan balik dari request yang diberikan. Dari diagram dapat dilihat aktifitas-aktifitas yang dapat dilakukan oleh jabatan yaitu *read* tabel surat masuk, *create* dan *read* item pada tabel surat\_keluar, *create* dan *read* item pada tabel surat\_keputusan, *create*, *read*, *update*, dan *delete* item pada tabel disposisi, dan *read* dan *update* kolom *delegated inbox* dan *delegated outbox* untuk memberi izin hak akses. Sedangkan untuk karyawan aktifitas yang dapat dilakukan yaitu *create* dan *read* item pada tabel surat\_keluar, dan *create* dan *read* item pada tabel surat\_keputusan jika memiliki izin hak akses *delegated outbox* , serta *read* tabel surat masuk, dan *create*, *read*, *update*, dan *delete* item pada tabel disposisi jika memiliki izin hak akses *delegated inbox*.

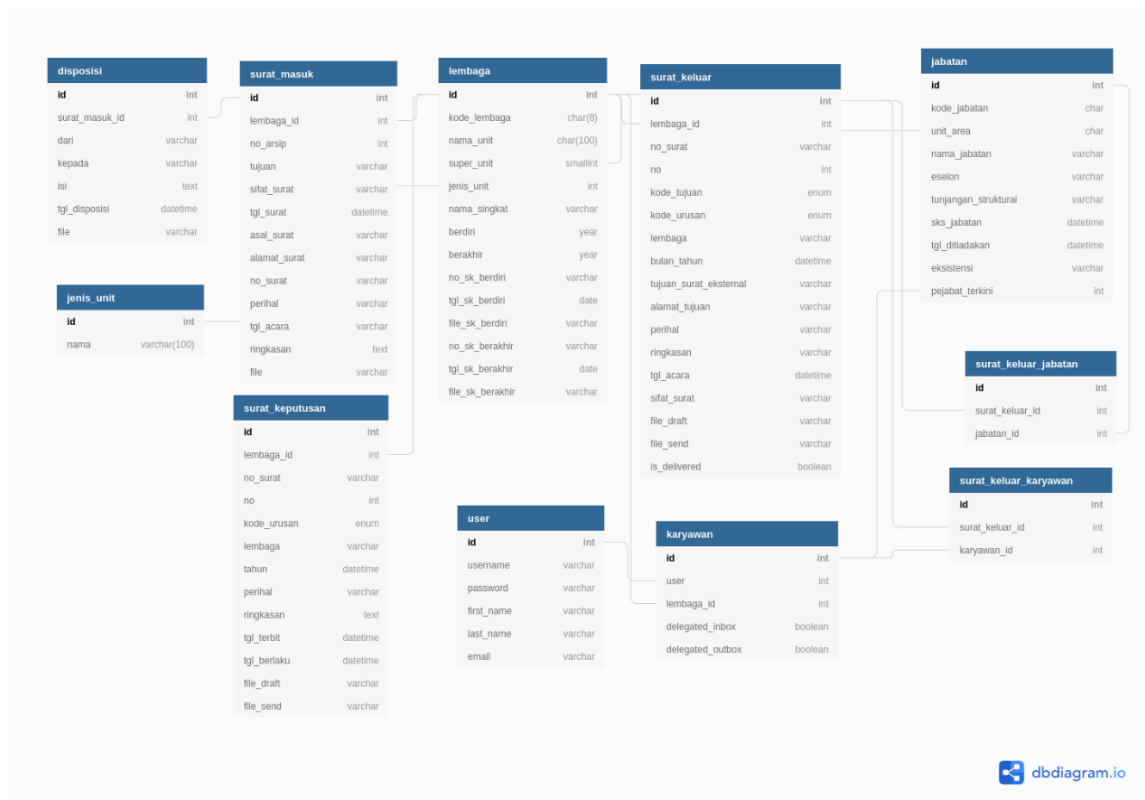


Gambar 3. Activity diagram jabatan



Gambar 4. Activity diagram karyawan

*Entity relationship diagram* digunakan untuk merancang dan memodelkan *database* (Yoshizumi, Kirishima, Goto, Tsuchida, & Yaku, 2017). Model *database* dari aplikasi pengelolaan surat ini ditunjukkan pada gambar 5.



Gambar 5. ERD aplikasi pengelolaan surat Universitas Muhammadiyah Surakarta

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Hasil

Hasil dari penelitian ini adalah sebuah *RESTful API* dengan alamat-alamat *URL* yang dapat digunakan untuk mengolah data aplikasi pengelolaan surat UMS. Untuk menggunakan *API* ini format yang diharuskan untuk mengirim data adalah *JSON*. Umpan balik dari sistem juga berupa data berformat *JSON*. Untuk proses autentikasi, terdapat 3 alamat *URL* seperti terlihat pada pada Tabel 1.

Tabel 1. Alamat-alamat *URL API* untuk proses autentikasi

No	Method	Alamat <i>URL</i>	Kegunaan
1	POST	<domain>/api/v1/token/	Memberi umpan balik berupa <i>access</i> dan <i>refresh token user</i>
2	POST	<domain>/api/v1/token/refresh/	Memberi umpan balik berupa <i>access token user</i>
3	POST	<domain>/api/v1/auth/login/	Memberi umpan balik berupa <i>access, refresh token</i> dan jabatan-jabatan <i>user</i>

Terdapat 2 *user* dalam pengelolaan surat ini, yaitu jabatan dan karyawan.

Jabatan dapat mengakses penuh pengelolaan surat lembaga tempat *user* berada, sedangkan karyawan hanya dapat mengakses pengelolaan surat jika memiliki izin hak akses.

*User yang ingin menggunakan API ini harus login terlebih dahulu dengan memasukkan username dan password pada alamat URL <domain>/api/v1/auth/login/. Jika berhasil login maka akan muncul access token, refresh token dan id-id lembaga tempat user menjabat jika user memiliki jabatan seperti ditampilkan pada gambar 6. Access token dapat digunakan untuk melakukan request pada alamat-alamat URL yang tertera Tabel 2 dan 3 sesuai hak akses user. Access token valid selama 5 menit, setelah itu user harus melakukan refresh token pada alamat URL <domain>/api/v1/token/refresh/ untuk mendapatkan access token baru yang juga valid selama 5 menit dengan memasukkan key refresh dan value berupa refresh token yang didapat ketika pertama kali login. Refresh token valid selama 24 jam, setelah itu user harus melakukan login lagi. Id lembaga dapat digunakan untuk memberi tahu API lembaga mana user akan masuk jika user mempunyai lebih dari 2 jabatan dengan cara memasukkan key lembaga dan value id lembaga tujuan pada header.*

Gambar 6. *Response* ketika *user* berhasil login

```
"refresh":
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2t1bl90eXBFIjoicmVmcmZzaCIsImV4cCI6MTU0NDSEODQyNywianRpIjoiOTQwNmRmMWMzMmYmNjNDY2MjgweYTEyMc2OGI2MTUxZWZlLjClc3VyX2kiJjoxMX0.1YL5bPZYWCByxBLNWNI776Dww-oIGNAMxn2dljqqtac",
"access":
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2t1bl90eXBFIjoieWNjZXNziiwiaXhwIjoxNTgwMTEyMzI3LjQdGKiOihYjYU4OWY4NzIyMWU0bzA4OAIzMWIxYzJkNThlZjgwOSIsInVzZXFjaXQ1byExfQ.RDoXjs0siPIRRimTcSIF_5dhmlwYP4evxaBdvj_Z96U",
"lembaga": [
    6
]
```

Untuk pengelolaan surat terdapat 12 alamat *URL* yang digunakan untuk mengelola surat masuk, keluar, dan keputusan seperti terlihat pada Tabel 2.

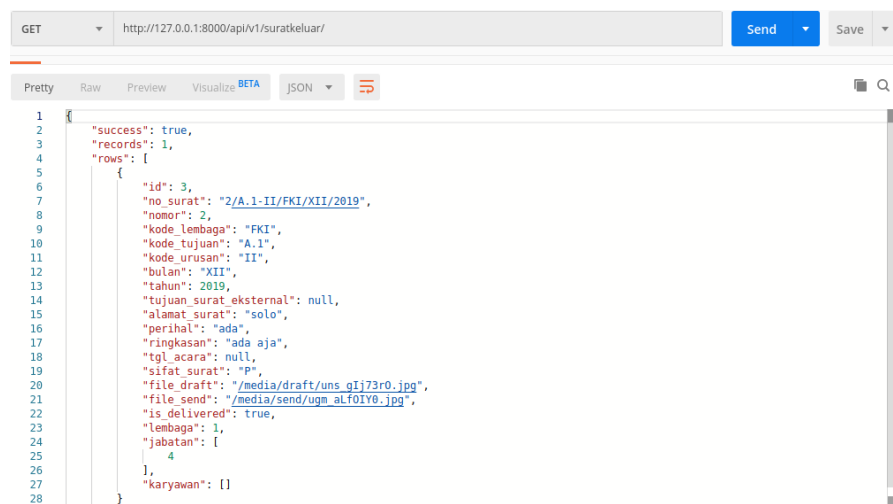
Tabel 2. Alamat-alamat *URL API* untuk mengelola surat

1	GET	<domain>/api/v1/suratkeluar/	Melihat daftar surat keluar yang sudah dibuat pada lembaga <i>user</i> berada
2	POST	<domain>/api/v1/suratkeluar/	Membuat surat keluar pada lembaga <i>user</i> berada
3	GET	<domain>/api/v1/suratkeluar/<id>/	Melihat surat keluar dengan id tertentu yang sudah dibuat pada lembaga <i>user</i> berada
4	PUT	<domain>/api/v1/suratkeluar/<id>/	Mengubah data surat keluar dengan <i>id</i> tertentu jika surat



			belum terkirim
5	DELETE	<domain>/api/v1/suratkeluar/<id>/	Menghapus surat keluar jika surat belum terkirim
6	GET	<domain>/api/v1/suratkeputusan/	Melihat daftar surat keputusan yang sudah dibuat pada lembaga <i>user</i> berada
7	POST	<domain>/api/v1/suratkeputusan/	Membuat surat keputusan pada lembaga <i>user</i> berada
8	GET	<domain>/api/v1/suratkeputusan/<id>/	Melihat surat keputusan dengan id tertentu yang sudah dibuat pada lembaga <i>user</i> berada
9	PUT	<domain>/api/v1/suratkeputusan/<id>/	Mengubah data surat keputusan dengan <i>id</i> tertentu jika surat belum terkirim
10	DELETE	<domain>/api/v1/suratkeputusan/<id>/	Menghapus surat keputusan jika surat belum terkirim
11	GET	<domain>/api/v1/suratmasuk/	Melihat daftar surat masuk yang sudah terkirim ke lembaga <i>user</i> berada
12	GET	<domain>/api/v1/suratmasuk/<id>/	Melihat surat masuk dengan id tertentu yang sudah terkirim ke lembaga <i>user</i> berada

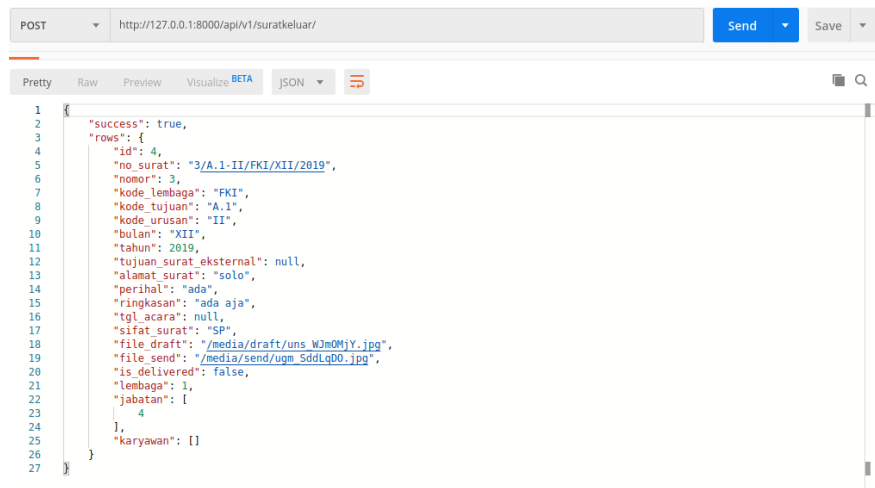
Setelah mendapatkan *token* dan memiliki izin hak akses, *user* dapat mengakses alamat-alamat *URL* pengelolaan surat. Pada gambar 7 menunjukkan *response* dari sistem jika menerima *request GET*.



Gambar 7. *Response* dari *request GET*

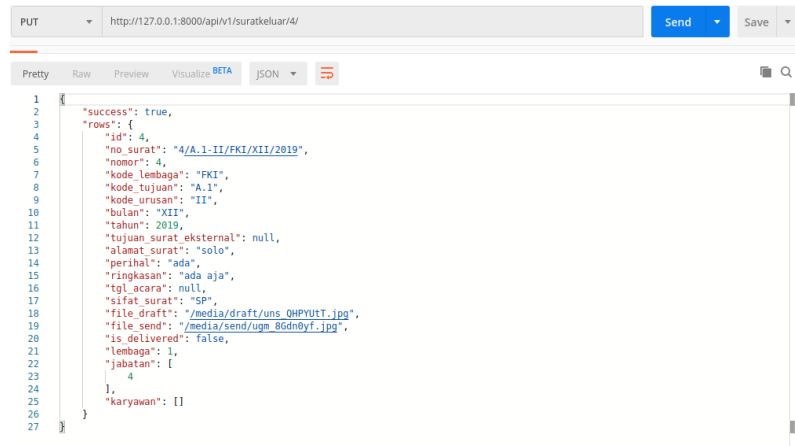
*Response* dari *GET* pada Tabel 2 akan mengembalikan data-data yang di-*request*, apakah itu data-data surat keluar, surat keputusan, atau surat masuk.

Untuk membuat surat keluar dan keputusan, dapat menggunakan method *POST*. Tujuan dari surat keluar ada 3, yaitu eksternal, karyawan, atau lembaga. Sedangkan tujuan surat keputusan ada 2 yaitu karyawan atau lembaga. Khusus untuk tujuan ke lembaga, surat yang sudah dikirim dapat langsung masuk ke surat masuk lembaga tujuan. Surat keluar dan surat keputusan dapat disimpan terlebih dahulu atau langsung dikirim. Jika ketika *request*, *key is\_delivered* bernilai *false*, maka surat akan tersimpan tetapi tidak akan langsung dikirim. Akan tetapi jika *key is\_delivered* bernilai *true*, maka surat keluar atau keputusan akan tersimpan dan akan masuk ke dalam surat masuk tujuan. Surat yang sudah terkirim tidak bisa untuk dihapus maupun diubah. Gambar 8 menunjukkan *response* yang diberikan jika *request POST* sukses. Jika berhasil akan memberikan *response* berupa data yang dimasukkan saat *request*.



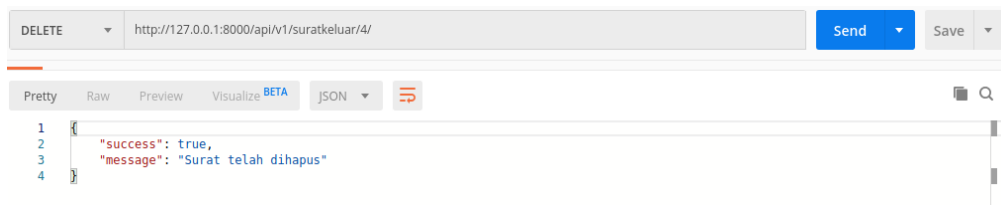
Gambar 8. *Response* dari *request POST*

Pada Tabel 2 Method *PUT* digunakan untuk mengubah surat keluar dan keputusan. Surat yang sudah dikirim tidak akan bisa diubah. Gambar 9 menunjukkan *response* dari *request PUT*. Jika berhasil akan memberikan *response* berupa data baru setelah perubahan.



Gambar 9. Response dari request PUT

Gambar 10 menunjukkan response ketika surat berhasil dihapus.



Gambar 10. Response dari request DELETE

Pada Tabel 3 menunjukkan alamat-alamat URL untuk melihat jabatan, karyawan, lembaga, serta melihat dan mengubah hak akses karyawan.

Tabel 3. Alamat-alamat URL API untuk jabatan, karyawan, lembaga dan hak akses

1	GET	<domain>/api/v1/lembaga/	Melihat daftar lembaga yang ada
2	GET	<domain>/api/v1/lembaga/<id>/	Melihat lembaga dengan id tertentu
3	GET	<domain>/api/v1/jabatan/	Melihat daftar jabatan yang ada
4	GET	<domain>/api/v1/jabatan/<id>/	Melihat jabatan dengan id tertentu
5	GET	<domain>/api/v1/karyawan/	Melihat daftar karyawan yang ada
6	GET	<domain>/api/v1/karyawan/<id>/	Melihat karyawan dengan id tertentu
7	GET	<domain>/api/v1/hak-akses/	Melihat hak akses karyawan pada lembaga pejabat berada
8	PUT	<domain>/api/v1/hak-akses/<id>/	Mengubah hak akses karyawan dengan id tertentu pada lembaga pejabat berada

Data karyawan, lembaga, dan jabatan pada URL akan mengambil dari web

service UMS. Sedangkan alamat *URL* hak akses dapat digunakan untuk mengelola hak akses karyawan. Hak akses karyawan dibagi menjadi 2 yaitu *delegated inbox* dan *delegated outbox*. *Delegated inbox* memberi hak akses karyawan untuk mengelola surat masuk dan disposisi. Sedangkan *delegated outbox* memberi hak akses karyawan untuk mengelola surat keluar dan surat keputusan. *Key* yang digunakan untuk *delegated inbox* adalah *delegated\_inbox* sedangkan untuk *delegated outbox* adalah *delegated\_outbox*. *Value true* pada *key* menunjukkan bahwa karyawan memiliki hak akses sedangkan *value false* menunjukkan karyawan tidak memiliki hak akses

### 3.2 Pengujian Black box

Pengujian blackbox adalah urutan pengujian yang berjalan untuk memeriksa apakah semua fungsi dalam sistem telah berfungsi dengan baik dan memenuhi harapan. Tabel 4 menampilkan hasil uji *black box API* pengelolaan surat UMS.

Tabel 4. Hasil uji *black box API* pengelolaan surat UMS

No	Method	Alamat <i>URL</i>	Input	Output	Hasil
1	POST	<domain>/api/v1/token/	<i>Username</i> dan <i>password</i>	<i>Token user</i>	Valid
2	POST	<domain>/api/v1/auth/login/	<i>Username</i> dan <i>password</i>	<i>Token</i> dan <i>id-id lembaga user</i>	Valid
3	POST	<domain>/api/v1/auth/logout/	-		Valid
4	GET	<domain>/api/v1/suratkeluar/	-	Data surat-surat keluar	Valid
5	POST	<domain>/api/v1/suratkeluar/	Data surat keluar baru	Data yang diinputkan	Valid
6	GET	<domain>/api/v1/suratkeluar/<id>/	-	Data surat keluar dengan id tertentu	Valid
7	PUT	<domain>/api/v1/suratkeluar/<id>/	Data perubahan surat keluar pada id tertentu	Perubahan data sesuai <i>input</i> -an baru	Valid
8	DELETE	<domain>/api/v1/suratkeluar/<id>/	-	Keterangan surat berhasil dihapus	Valid
9	GET	<domain>/api/v1/suratkeputusan/	-	Data surat-surat keputusan	Valid
10	POST	<domain>/api/v1/suratkeputusan/	Data surat keputusan	Data yang diinputkan	Valid

			baru		
11	GET	<domain>/api/v1/suratkeputusan/<id>/	-	Data surat keputusan dengan id tertentu	Valid
12	PUT	<domain>/api/v1/suratkeputusan/<id>/	Data perubahan surat keputusan pada id tertentu	Perubahan data sesuai <i>input</i> -an baru	Valid
13	DELETE	<domain>/api/v1/suratkeputusan/<id>/	-	Keterangan surat berhasil dihapus	Valid
14	GET	<domain>/api/v1/suratmasuk/	-	Data surat-surat masuk	Valid
15	GET	<domain>/api/v1/suratmasuk/<id>/	-	Data surat masuk dengan <i>id</i> tertentu	Valid
16	GET	<domain>/api/v1/lembaga/	-	Data lembaga-lembaga	Valid
17	GET	<domain>/api/v1/lembaga/<id>/	-	Data lembaga dengan id tertentu	Valid
18	GET	<domain>/api/v1/jabatan/	-	Data jabatan-jabatan	Valid
19	GET	<domain>/api/v1/jabatan/<id>/	-	Data jabatan dengan id tertentu	Valid
20	GET	<domain>/api/v1/karyawan/	-	Data karyawan-karyawan	Valid
21	GET	<domain>/api/v1/karyawan/<id>/	-	Data karyawan dengan id tertentu	Valid
22	GET	<domain>/api/v1/hak-akses/	-	Data karyawan-karyawan dengan hak aksesnya	Valid
23	PUT	<domain>/api/v1/hak-akses/<id>/	Data hak akses pada id tertentu	Perubahan data sesuai <i>input</i> -an baru	Valid

#### 4. PENUTUP

*API* pengelolaan surat UMS sudah dibuat dan bekerja dengan baik, serta sudah memenuhi ekspektasi dalam hal fungsionalitas. Namun *API* ini masih perlu disinkronkan dengan web service UMS agar dapat diimplementasikan. Untuk kedepannya, *API* ini dapat disinkronkan dengan *web service* UMS yang selanjutnya dapat dikonsumsi ke dalam *website* atau aplikasi *mobile*.

## DAFTAR PUSTAKA

- Adithela, S. P., Christie, M., Marru, S., & Pierce, M. (2018). Django content management system evaluation and integration with apache airavata. In *Proceedings of the Practice and Experience on Advanced Research Computing* (pp. 1-4).
- Hagberg, J., Sundström, M., & Nicklas, E. Z. (2016). The digitalization of retailing: an exploratory framework. *International Journal of Retail & Distribution Management*, 44(7), 694-712.
- Isaias, P., & Issa, T. (2015). *High level models and methodologies for information systems* (pp. 1-145). New York, NW: Springer.
- Parviainen, P., Tihinen, M., Kääriäinen, J., & Teppola, S. (2017). Tackling the digitalization challenge: how to benefit from digitalization in practice. *International journal of information systems and project management*, 5(1), 63-77.
- Supriyana, A., Azmin, G. G., Nurriyani, R., & Rahmawati, A. (2015). Pelatihan Penggunaan Ejaan Yang Disempurnakan Dan Kalimat Efektif Pada Penulisan Surat Resmi Bagi Guru Sekolah Dasar Di Jakarta Timur. *Sarwahita*, 12(1), 5-10.
- Iqbal, S. Z., & Idrees, M. (2017). Z-SDLC Model A New Model For Software Development Life Cycle (SDLC). *International Journal of Engineering and Advanced Research Technology (IJEART)*, 3(2), 8.
- Herlambang, R., & Thamrin, H. (2018). Aplikasi Paperless Library dan Pengukuran Dampak dengan Model IS-IMPACT. *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, 4(2), 69-76.
- Yoshizumi, T., Kirishima, T., Goto, T., Tsuchida, K., & Yaku, T. (2016, July). A graph grammar for entity relationship diagrams. In *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)* (pp. 810-815). IEEE.